tion converts it into a modified document of the higher version and sends the modified document of the higher version to the user using the higher version, the apparatus **200** according to the present invention is also suitable for another scenario, wherein a user using the application of a lower version sends a document of the lower version to the apparatus **200** of the present invention at a network server, and the apparatus **200** of the present invention converts it into a document of a higher version, and sends it to a user using the application of a higher version, who then sends a modified document of the higher version to the apparatus **200** of the present invention, and the apparatus **200** of the present invention converts it into a modified document of the lower version and sends the modified document of the lower version to the user using the lower version.

[0075] According to another embodiment of the present invention, the conversion module **202** for converting between documents of different versions of an application by using a conversion stack **201** is further configured for: obtaining a document of a first version; successively converting the document of the first version into zero, one or more documents of intermediate versions between the first version and the second version and a document of the second version by using one or more differentiation models between data models between the first version and the second version in the conversion stack; and providing the document of the second version.

[0076] Providing the document of the second version may include, for example, sending the document through a network or providing it in other manners to a user who uses the application of the second version, so that the user can check or modify it, or directly displaying the document on a user interface of the application of the second version, and the like.

[0077] The above describes the apparatus **200** for providing inter-version document compatibility according to an embodiment of the present invention. It should be pointed out that the above description is only exemplary, rather than limiting to the scope of the present invention. Compared with what is described, the apparatus **200** of the present invention can have more, fewer or different modules and the connection and containment relationships between the modules can be different. Part of or all the functions of each of the above modules can also be accomplished by other existing or new modules.

[0078] For example, according to some embodiments of the present invention, the apparatus **200** may exclude one or more of the above optional modules; the apparatus **200** can further include a receiving module specialized for sending and receiving documents, and so on. All these variations are within the spirit and scope of the present invention.

[0079] The method for providing inter-version document compatibility can be executed according to an embodiment of the present invention by the apparatus **200** for providing inter-version document compatibility according to an embodiment of the present invention or any other device. For brevity, part of the details repetitive with the above description is omitted in the following description. Therefore, a more detailed understanding of the method of the present invention can be obtained by referring to the above description.

[0080] FIG. **8** shows a method for providing inter-version document compatibility according to an embodiment of the present invention. As shown in the figure, the method includes following steps.

[0081] In step **801**, a conversion stack is provided, and the conversion stack includes differential models between data models of different versions of an application.

[0082] In step **802**, the conversion stack is used to convert between documents of different versions of the application, so as to provide compatibility between documents of different versions of the application.

[0083] According to an embodiment of the present invention, the differentiation models between data models of the different versions of the application are differentiation models between the data models of adjacent versions of the application.

[0084] According to an embodiment of the present invention, the conversion stack is in the application of each version, and includes differentiation models between the data models of the version and lower versions, and the conversion is performed in the application of the version.

[0085] According to an embodiment of the present invention, the conversion stack is at a network server, and the conversion is executed as a networked service at the network server.

[0086] According to an embodiment of the present invention, the step **801** for providing the conversion stack includes the following sub-steps: obtaining data models of different versions of the application; and generating differentiation models between the data models of different versions of the application according to the data models of different versions of the application, and storing the differentiation models to the conversion stack.

[0087] Now referring to FIG. **9**, it shows the sub-steps included in the step **802** for converting between documents of different versions of an application by using the conversion stack according to an embodiment of the present invention.

[0088] As shown, a document of a higher version is obtained in step **901**.

[0089] In step **902**, the document of the higher version is successively converted into zero, one or more documents of intermediate versions between the higher version and the lower version and the document of the lower version by using one or more differentiation models between data models between the higher version and the lower version in the conversion stack, and the document of the higher version and the zero, one or more documents of intermediate versions between the higher version and the lower version or the relevant information therein are stored.

[0090] In step **903**, the document of the lower version is sent to the application of the lower version, so as to require modifications to the document of the lower version may be performed by using the lower version of the application.

[0091] In step **904**, a modified document of the lower version is received from the application of the lower version.

[0092] In step **905**, the modified document of the lower version is converted into zero, one or more modified documents of intermediate versions and a modified document of the higher version by using the one or more differentiation models between the data models between the higher version and the lower version in the conversion stack and by successively merging with the zero, one or more stored documents of intermediate versions between the higher version and the lower version and the stored document of the higher version or the relevant information therein.

[0093] Now referring to FIG. **10**, it shows the sub-steps included in the step **802** for converting between documents of